McGINN & GIBB, P.C.
A PROFESSIONAL LIMITED LIABILITY COMPANY
PATENTS, TRADEMARKS, COPYRIGHTS, AND INTELLECTUAL PROPERTY LAW
1701 CLARENDON BOULEVARD, SUITE 100
ARLINGTON, VIRGINIA 22209
TELEPHONE (703) 294-6699
FACSIMILE (703) 294-6696

APPLICATION
FOR
UNITED STATES
LETTERS PATENT

APPLICANTS: Richard Harold Boivie, Daniel Manuel Dias, Colin George Harrison, Eric M. Levy-Abegnoli, Jean A. Lorrain, Kiyoshi Maruyama, and Pascal Pol Marie Thubert

FOR: SYSTEM AND METHOD FOR ENFORCING COMMUNICATIONS BANDWIDTH BASED SERVICE LEVEL AGREEMENTS TO PLURALITY OF CUSTOMERS HOSTED ON A CLUSTERED WEB SERVER

DOCKET NO.: YO999-374

SYSTEM AND METHOD FOR ENFORCING COMMUNICATIONS BANDWIDTH BASED SERVICE LEVEL AGREEMENTS TO PLURALITY OF CUSTOMERS HOSTED ON A CLUSTERED WEB SERVER

Field of the Invention

5 The present invention relates generally to a world-wide network, and more particularly to those of Internet World Wide Web (WWW) sites of various owners hosted by a service provider using a cluster or group of servers and meeting with agreed upon service levels.

Background of the Invention

 The Internet is the world's largest network, and it has become essential in organizations

10 such as government, academia, small, medium and large businesses as well as to individual consumers. Many businesses have started out-sourcing their electronic business ("e-business") and electronic commerce ("e-commerce") Web sites to service providers instead of running their Web sites on their own server(s) and managing them by themselves.

 Such a service provider needs to install a collection of servers (termed a Web Server

15 Farm (WSF), Universal Server Farm (USF), or Web Server Cluster) which can be used by many different businesses to support their e-commerce and e-business. These business customers (e.g., the service provider's "customers") have different "capacity" requirements for their Web sites. The users of e-commerce (consumers, business partners, etc.) access the Web Server Farm by

first logging into the Internet. The Web Server Farm (WSF) is connected to the Internet via high speed communications links such as T3 and OCx links. These links are shared by all the Web sites, and all the users who are accessing the services hosted by the Web Server Farm.

When businesses out-source their e-commerce and/or e-business to a service provider, they must get some guarantee on the services they are getting from the service provider for their sites. Once the service provider makes a commitment to a customer to provide a certain "level" of service (e.g., termed a Service Level Agreement (SLA)), the provider must guarantee that level of service to that customer.

A general service level agreement (SLA) on communications link bandwidth usage for a customer can be denoted by a pair of bandwidth constraints: the minimum guaranteed bandwidth $B(i, min)$ and the maximum bandwidth bound $B(i, max)$ for each i-th customer.

The minimum (or min) bandwidth $B(i, min)$ is a guaranteed bandwidth that the i-th customer will receive regardless of the bandwidth usage by other customers. The maximum (or max) bandwidth $B(i, max)$ is an upper bound on the bandwidth that the i-th customer may receive provided that some unused bandwidth is available (e.g., bandwidth not being currently used by other customers).

Therefore, the range between $B(i, min)$ and $B(i, max)$ represents bandwidth provided on an "available" or "best-effort" basis to a customer, and it is not necessarily guaranteed that the customer will obtain this bandwidth.

Generally, the unit cost to use the bandwidth up to $B(i, min)$ is less than or equal to the unit cost to use the bandwidth between $B(i, min)$ and $B(i, max)$. Such a unit cost assigned to one customer may differ from those assigned to other customers.

YO999-374

2

In the environment of Web site hosting, where communications link(s) between the Internet and a server farm is shared by a number of customers (i.e., traffic to/and from the customer Web sites are sharing the communications link(s)), the bandwidth management on the outbound link (i.e., the link from a server farm to the Internet) is more important than the

5 bandwidth management on the inbound link since the amount of traffic on the outbound link is several orders of magnitude greater than that on the inbound link.

Furthermore, in most cases, the inbound traffic to the server farm is directly responsible for the outbound traffic generated by the server farm. Therefore, the service level agreements $(B(i, min), B(i, max))$ are generally applied on the outbound link bandwidth usage.

10 A conventional method for leaky bucket traffic shaping uses fair queuing collision arbitration. This method uses a set of queues and virtual finishing time for scheduling Asynchronous Transfer Mode (ATM) cell traffic. The method could be implemented to regulate packet traffic to enforce the minimum bandwidth SLA, $B(i, min)$ to each customer.

Another conventional method for shaping traffic in a packet-switched network uses a set

15 of queues and the conformance time to shape packet traffic. The method can be used to regulate packet traffic to enforce the minimum bandwidth SLA, $B(i, min)$ to each customer.

Some commercial products (e.g., Xedia Access Point (www.Xedia.com) and Phasecom's SpeedClass (www.speed-demon.com) can be used to regulate directly the outbound traffic to meet with the (minimum, maximum) bandwidth SLA for each customer.

20 However, while the conventional systems and methods such as those mentioned above, could be reasonably applied to enforce service level agreements (SLAs) on the outbound link

usage by each customer (and on customer traffic class or type), some can only support the minimum bandwidth SLA and some can support the (minimum, maximum) bandwidth SLA.

A major problem of these systems and methods is that they enforce the outbound bandwidth SLA by throttling (i.e., dropping some of) the traffic already generated by specific

5   source (IP) addresses.

A major problem arises when some packets must be dropped because either the outbound link has been congested for a sustained period of time so that some queues are running out of space, and/or a particular customer's outbound link usage has been exceeding the maximum bandwidth SLA and its queue(s) has filled up. That is, when a queue has filled up,

10   packets coming to that queue are randomly dropped. This means the packet-dropping affects many responses from a server farm since each response consists of many packets. This dropping of packets triggers many Transmission Control Protocol (TCP) retransmission, which leads to even further retransmission of packets for error recovery, causing thrashing and then eventually leading to a slowdown of many connections or disconnection of connections.

15   Another problem of dropping outbound packets is wasting server resources in generating responses which cannot be delivered to end users (i.e., servers are kept busy for generating undeliverable responses, yet they could have been used for generating responses for customers whose outbound bandwidth usage are below the minimum guarantees).

Yet another problem is, classification of outbound traffic is limited since the outbound

20   packet does not say for which type request the packet was generated. This will limit the degree to which differentiated services could be applied in controlling bandwidth usage.

SUMMARY OF THE INVENTION

In view of the foregoing and other problems, disadvantages and drawbacks of the conventional systems and methods, an object of the present invention is a system and method to guarantee and deliver the (minimum, maximum) based communications link bandwidth Service

5   Level Agreements (SLAs) to customers whose applications (e.g., Web sites) are hosted by a server or a server cluster.

Another object of the present invention is to prevent any single customer's traffic in the server or server cluster from monopolizing (e.g., "hogging") the entire bandwidth resource and penalizing others.

10   Yet another object of the present invention is to provide a feedback method and system for enforcing the outbound link bandwidth SLAs by throttling and shaping the inbound traffic to a server or a server cluster.

A further object of the present invention is to provide differentiated services to various types of traffic within a defined class of traffic by supporting multilevel selective- packet-drop

15   criteria.

A still further object of the present invention is to serve the incoming requests by priority order and not necessarily in their order of arrival.

Yet another object of the present invention is to provide a method and system for classifying and shaping traffic using various levels from the Internet Protocol (IP) addresses to

20   User Resource Locators (URLs).

YO999-374

5

Another object of the invention is to provide a method and system for maximizing the usage of the outbound link bandwidth and to provide fair usage to various classes of traffic.

Yet another object of the present invention is to provide a method and system for maximizing the potential service revenue in allocating unused bandwidth to various classes of

5   traffic.

A still further object of the invention is to provide a method and system for avoiding the generation of output from a server or a server cluster if that output could not be delivered to end users.

Another object of the present invention is to provide a method and system for dropping

10   requests and/or connections to minimize the negative business impact to customers.

In a first aspect of the present invention, a system and method are provided to guarantee and deliver the communications link bandwidth-based Service Level Agreements (SLAs) to customers whose applications (e.g., Web sites) are hosted by a server or a server farm.   A method of controlling and guaranteeing a service level agreement (SLA), based on a

15   communications outbound link bandwidth usage to a plurality of customers having electronic business activity hosted by at least one server as a server farm, includes a) monitoring the outbound communications bandwidth usage by each customer traffic to determine a level of service being provided to each customer with respect to the agreed service level agreement in each service cycle time per unit of time; and b) controlling a flow of incoming requests to each

20   customer business activity application so as to guarantee a level of service previously agreed to the customer by queuing requests and by selectively dropping requests to the customer to guarantee the agreed service levels to the customer. The controlling step controls and guarantees

each outbound link usage based service level agreement by controlling the flow of incoming requests to the at least one server.

The Service Level Agreements preferably are in the form of (minimum, maximum) bandwidth bounds where the minimum bandwidth represents the guaranteed bandwidth while the maximum bandwidth represents the upper bound to the as-available use bandwidth. A customer may have an SLA for each type traffic. The system enforces the outbound link bandwidth SLAs by throttling and shaping inbound traffic to a server farm.

Preferably, incoming traffic is classified into various class/type queues. Each queue has multiple water-levels to trigger selective-dropping of packets prior to dropping any new arriving packets. Traffic in queues is scheduled and shipped to a server farm using a real-time scheduling algorithm that support and enforce the (minimum, maximum) bandwidth-based SLA to each customer traffic. The system uses the information on the outbound link bandwidth usage by various class traffic as a primary means to throttle and shape inbound traffic to a server farm. The scheduling algorithm looks for an opportunity to increase the revenue to a server farm operator when allocating bandwidth resources beyond the minimum bandwidth SLAs.

With the unique and unobvious structure and method of the invention, the (minimum, maximum) based communications link bandwidth SLAs are guaranteed and delivered to customers whose applications are hosted by a server or a server cluster. Further, single customer traffic (a class traffic belonging to a single customer) in the server or server cluster are prevented from monopolizing an entire bandwidth resource at the expense of other customers. Along these lines, the usage of the outbound link bandwidth is maximized and yet fair usage is provided to

YO999-374

7

various classes of traffic. Additionally, feedback is provided for enforcing the outbound link

bandwidth SLAs by throttling and shaping the inbound traffic to a server or a server cluster.

Additionally, with the present invention, differentiated services are provided to various

types of traffic within a defined class of traffic by supporting multilevel selective- packet-drop

5   criteria. Further, incoming requests are served by priority order and not necessarily in their order

of arrival. Additionally, traffic is classified and shaped using various levels such as from the

Internet Protocol (IP) addresses to User Resource Locators (URLs). The potential service

revenue is maximized in allocating unused bandwidth to various classes of traffic. Further, the

output generation from a server or a server cluster is avoided if that output cannot be delivered to

10   end users. Thus, server resources are conserved, and unnecessary output (packets) are not

generated and subsequently dropped. Further, requests and/or connections are dropped to

minimize the negative business impact to customers.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from

15   the following detailed description of a preferred embodiment of the invention with reference to

the drawings, in which:

Figure 1 illustrates an overall system and environment of the present invention;

Figure 2 illustrates a communications bandwidth manager 110 according to the present

invention for managing and monitoring a communications bandwidth on an output data path;

YO999-374

8

Figure 3 illustrates the components of the Communications Bandwidth Manager 110 including a traffic estimator 340, a scheduler 320, and a queuing system 360;

Figure 4A illustrates a flowchart of an inbound traffic scheduling method 400; and

Figure 4B illustrates a method of computing b(i), the expected bandwidth usage per a

5   request that is admitted to a server farm.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

OF THE PRESENT INVENTION

Referring now to the drawings, and more particularly to Figures 1-7, an embodiment of the present invention will be described herein below.

10   Figure 1 shows the overall system 100 and environment for the present invention. As shown in Figure 1, clients 130 (e.g., Client_1, Client_2, Client_3, ..., Client_n) on a world-wide network (e.g., the Internet) 120, typically running Web browsers, make requests for web data located on a cluster of Web servers 100.

The requests arrive at a Communications Bandwidth Manager (CBM) 110. The CBM

15   110 presents a single IP address to the Web server cluster. The CBM 110 may queue incoming requests in a set of queues (as described in further detail below) in order to satisfy service level agreements (also described below). The CBM selects a request from the queues, selects one of the servers 101 (e.g., Server_1, Server_2, Sever_3, etc.) to service the request, and sends the request to that server.

20   The selected server receives the Web request, services it, and sends the response directly back to the client 130 (e.g., Client_3) along path 160. The return path 160 between the servers

YO999-374

9

100 and the Internet 120, is shared by the cluster of servers. This path 160 carries the bulk of the

Web server bandwidth load.

The present invention is directed at controlling the allocation of this link bandwidth

among multiple customer Web sites hosted on the Web server cluster 100, by controlling

5   incoming Web requests at the CBM 110.

Figure 2 illustrates the main concept of the present invention embodied in the

Communications Bandwidth Manager (CBM) 110. The CBM 110 receives incoming (i.e.,

inbound) Web requests 240 from Web clients 130, on input link 241.

The input requests 240 are queued in a set of queues 250 (e.g., in order of arrival time)

10   The requests in a queue 250 are serviced first-in, first out (FIFO) within each particular class

(type) of requests. While Figure 2 only shows a single queue 250, there exists queues for each

traffic class for each customer. That is, a customer may have a single queue or multiple queues.

If there are multiple queues, there are queues for each traffic type (e.g., class, priority, etc.).

Thus, within each of the classes, the requests are serviced in a FIFO manner. Once, the request

15   arrives in FIFO, the request is placed into a predetermined queue based on the customer, different

customer's Web site, server, etc. Hence, within each particular queue, the FIFO processing is

preserved but not across different queues.

In addition to Web input requests, the input requests 240 include packets associated with

the ongoing Web requests, such as acknowledgment packets.

20   The acknowledgment packets contain information on the quantity of outbound data that is

being acknowledged, as specified in the conventional protocols, such as the specification of the

TCP/IP (Transmission Control Protocol / Internet Protocol). This information may be used to

estimate the volume of data that was sent out on the outgoing path 160 from the server 101.

As illustrated in Figure 2, the outgoing path 160 may be monitored, in place of

monitoring acknowledgment packets (e.g., generated by the server and indicating, for example,

5   the number of data bytes delivered/received by a customer).

The path 160 carries the outbound traffic (i.e., responses to requests). Hence,

monitoring the output on the outgoing data path 160 can be monitored and it can be observed

how much output is being generated. If the amount of bandwidth being used by a particular

customer exceeds the number of bandwidth per the SLA, then feedback can be generated to

10   reduce the number of inbound requests being accepted.

Thus, the present invention provides a feedback loop, so as to provide monitored data 260

indicating the bandwidth on output path 160 being used by each of the Web sites hosted on the

Web server cluster 100.

This monitored data 260 may be obtained directly by monitoring the output path 160

15   (e.g., monitoring the flow and the available bandwidth) at each of the Web servers in the cluster.

Based on the monitored data 260 (or the data contained in acknowledgment packets),

configuration information (e.g., the connectivity between the CBM and servers) and

accumulated information (e.g., server performance data) based on operation of the cluster, the

CBM 110 selects from the queued incoming requests 240 (e.g., based on the exemplary

20   scheduling processes shown in Figures 4A and 5A which describe the criteria for selecting

certain requests over others), selects a Web server node 101 (e.g., based on a determination made

by a conventional load balancer or the like which selects one node over another) to service the request, and forwards the request 280 to the selected server 101.

Based on the load of the outgoing path 160, some incoming requests may accumulate in queues 250. Thus, if traffic is heavy (e.g., above a predetermined threshold), some or all of the queues 250 may overflow, or some incoming requests may get delayed for a time period beyond a threshold. In these cases, the CBM 110 selects one or more incoming requests and discards them. Hence, once the queue is backed up, then there may be a delay of the incoming requests and a discarding of requests. Upon the queue becoming full, the incoming requests coming to a particular queue may be discarded in order of arrival. It is noted that the scheduler determines whether to admit (or not admit) a request. Typically, requests in a queue will not be dropped, but only those request newly-arriving will be dropped.

Alternatively, for these "to-be discarded" requests, the CBM may either directly, or via a selected server, return a message to the Web client 130 that the server complex is overloaded. There may be a dedicated server to return such a message (e.g., not to process requests, but only to return such "Server Busy" messages). Once, for example, a browser receives such a message, the browser can proceed to a next stage. Otherwise, the browser must wait until a "time out" message is received. After the "time out" the browser would move to a next process.

For example, in an Internet browser, when "CNN" is clicked, the "CNN" server will bring up a template, and within the template there are instructions to the Internet browser to "click" on various fields so that the template can be filled with content (e.g., images, text, etc.). Such "clicks" by the browser are made transparent to the user, to obtain the plurality of information. When the browser cannot obtain something, then the browser must retransmit the

YO999-374

12

request. However, to avoid this, the dedicated server described above could be employed, thereby freeing the browser for other tasks.

Figure 3 shows the components of the CBM 110. Incoming Web requests on input link 241 are placed in the queuing system 360. Monitored data 260 on the usage of the outgoing data path 160 arrive, and are gathered, at the traffic estimator 340. The traffic estimator may also gather the output load information from acknowledgment packets that arrive at the queuing system 360, via link 310.

The traffic estimator compiles the load information, and inputs the composed information to a scheduler component 350 over link 320. Based on the data from the traffic estimator, service level agreement (SLA) information (e.g., provided by the server(s)), and other configuration information (e.g., the connectivity between the CBM and servers), the scheduler 350 selects a request in the queuing system 360, and determines the server node 101 to service the request, and sends the selected request to the server node over link 280.

The scheduler 350 also determines if requests or other associated packets in the queuing system 360 are to be discarded, or whether a direct response indicating an overloaded condition is to be sent to the client Web browser.

Exemplary Scheduling Method

Figure 4A shows the scheduling method/algorithm 400 for inbound requests to a server farm after the requests have been queued in various queues 360.

The scheduling method 400 starts at step 401. The rest of steps (e.g., 402 through 409) will be repeated in a preset service cycle time. Step 402 resets control variables such as $b(i)$, B,

B(i) and R(i) for all class *i*. Here, *b(i)* represents the expected outbound link bandwidth usage in a service cycle time had a request in the class *i* been admitted to the server farm, *B* represents the current available bandwidth space that is set to *Bmax* at the beginning of each service cycle time (e.g., *Bmax* is the maximum usable bandwidth available on the outbound link in a service cycle time), *B(i)* represents the expected bandwidth usage had *R(i)* number of requests been dispatched to a server farm and *R(i)* represents the number of requests scheduled (and dispatched) thus far within the cycle time.

Step 403 determines whether or not a given service cycle time has expired. If it has expired (e.g., "YES"), then the method loops back to step 402. If the given service cycle time has not yet expired (e.g., "NO"), then step 404 checks whether or not any bandwidth is left for use. If no space is available (i.e., *B* is less-than-or-equal-to 0, denoted by *B <LE> 0*), step 405 is executed, and then goes back to step 402.

If some bandwidth space is available (i.e., *B <LT> 0, B* greater than zero), then the algorithm executes 406 and checks if there is any queue *i* such that a request is in that queue and its planned bandwidth usage is less than (<LT>) its minimum bandwidth SLA, B(i, min).

Step 406 guarantees the minimum bandwidth SLA. If the condition of the step 406 is "YES", then the algorithm schedules (and dispatches) one request from queue *i* and updates control variables; *B(i) = B(i) + b(i), R(i) = R(i) + 1, and B = B - b(i)*, and then goes back to step 403.

If the condition in step 406 is "NO" (i.e., as of this moment, every class has received the minimum bandwidth SLA service), then step 407 is executed, in which the algorithm determines

whether there is class i such that there is a request in that queue and its bandwidth usage is less than its maximum bandwidth SLA, B(i, max).

If the condition in step 407 is "NO", then the method loops back to step 403. If the condition 407 is "YES", then step 408 is executed and class i is selected. The selection of a class

5   is based on any given criteria predetermined by the designer. Examples of these criteria are priorities, weights assigned to classes, which class pays more than others, which class should receive preferential treatment over others, etc.

Once a class is selected in step 408, step 409 is executed to update the control variables, and then loops back to step 403. Steps 406, 407 and 408 provide the best effort services defined

10   by the range between the minimum bandwidth SLA, B(i, min) and the maximum bandwidth

SLA, B(i, max).

That is, for purposes of the invention, "best effort services" means that for each customer there is an SLA. Within the SLA, the customer is typically guaranteed a minimum bandwidth, or rate (bytes/bits per second) generated and delivered by their server. Thus, there is a minimum

15   guaranteed level of service (e.g., 1 MBits per second guaranteed). There also typically is a range of additional between the minimum guarantee and a maximum bandwidth which the customer may be offered (e.g., 2 Mbits per second for a total of 3 Mbits per second), to service additional users coming into the customer's Web server and thus the Web server may generate a lot more than 1 Mbit per second of traffic. The range between 1 Mbit per second (minimum bandwidth)

20   and 3 Mbit per second (maximum) is termed the "best efforts" range (also the "as available" range). When the customer starts using the bandwidth in this range, the customer is charged for

such usage. This bandwidth beyond the minimum is not guaranteed and the customer may not get it, depending upon other customer's usage and SLAs.

Figure 4B describes how b(i) in step 402 of Figure 4A can be computed. b(i) is derived by dividing Ba(i) by Ra(i) where Ba(i) represents the averaged/smoothed bandwidth usage that is derived from the measured or monitored bandwidth usage Bm(i) in the immediately preceding cycle time, and Ra(i) represents the averaged/smoothed number of requests.

Figure 4B shows one way of smoothing by using a constant "a" where a is a value between zero and one. The averaging/smoothing is used to absorb "spikes" (e.g., mitigate the effect of spikes on the waveform) to derive statistically meaningful values. There are many other conventional means to obtain the averaged or smoothed value as are known by one of ordinary skill in the art .

The scheduling method described above and shown in Figure 4A attempts to utilize the currently- available bandwidth as much as possible.

The method 400 can be modified easily to be applied to control and regulate the outbound link bandwidth usage directly to meet the (min, max) SLAs, similar approach used by the conventional products. Such a modification is related to the computation of b(i) in Step 402 in Figure 4A and in Figure 4B. Instead of "computing" b(i), the value b(i) can be "obtained" from the packet that is being scheduled and dispatched. (i.e., each packet carries its length information, and thus this packet length information can be used as b(i) in stead of computing b(i) as described in Figure 4b) This is possible since the method is applied directly to the traffic on the outbound link from the server farm to the Internet.

The CBM described in this invention can be implemented by hardware or software or both. For example, the CBM may be implemented by operating a computer, as embodied by a digital data processing apparatus, to execute a sequence of machine-readable instructions. These instructions may reside in various types of programmable signal-bearing media.

5   Thus, this aspect of the present invention is directed to a programmed product, including signal-bearing media tangibly embodying a program of machine-readable instructions executable by a digital data processor to perform the above method.

Hence, in addition to the hardware and process environment described above, a different aspect of the invention includes a computer-implemented method according to the present

10   invention, as described above, for enforcing communications bandwidth based on service level agreements to a plurality of customer hosted on a clustered web server, as described above. As an example, this method may be implemented in the particular hardware environment by executing a sequence of machine-readable instructions in the signal-bearing media.

With the present invention, a closed-loop control is provided whereby the outbound

15   traffic (e.g., responses) provides the control of the incoming traffic (e.g., requests).

While the invention has been described in terms of preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.